Contiene il pratico MASTER IN ACETATO

ELETTRONICA

N. 56 FEBBRAIO '90

L. 6000 - Frs. 9,00

Realizzazioni pratiche • TV Service • Radiantistica • Computer hardware

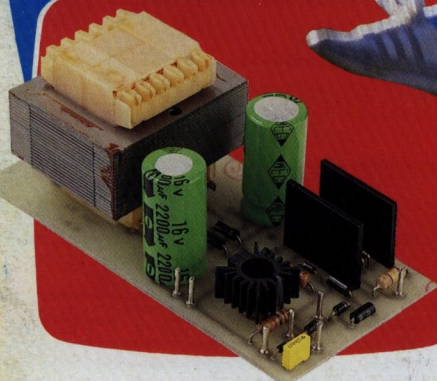
REALIZZAZIONI PRATICHE

Regolatore per caricabatterie

Chrono/impulse meter

COMPUTER HARDWARE

Mouse per C64

Alimentatore per EPROM

RADIANTISTICA

Ricevitore aeronautico

TACHIMETRO PER BICICLETTE

00 KM/H

BORSARI GIOVANNI

IN COLLABORAZIONE CON ETI ELECTRONICS TODAY INTERNATIONAL

TV SERVICE

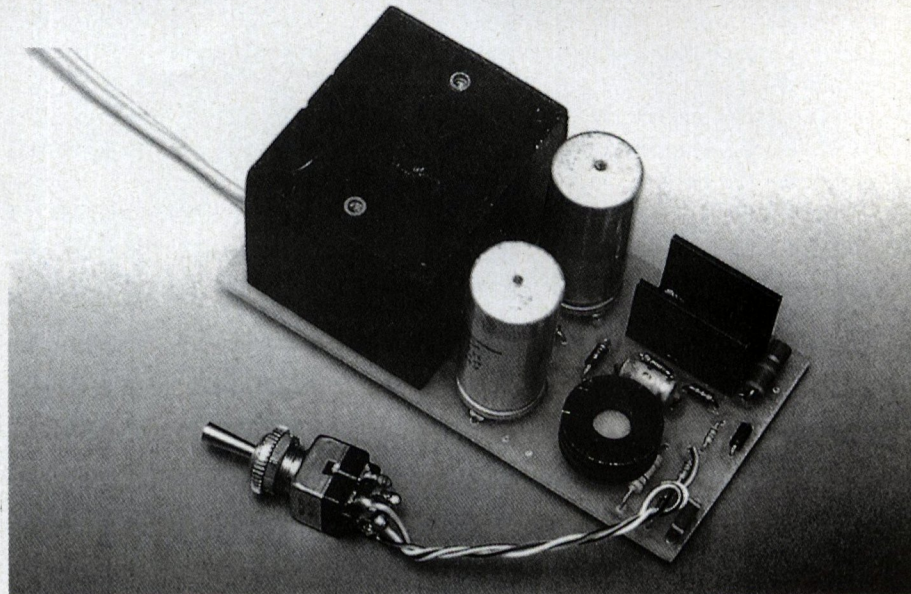
Saba C67S77 - II parte

Taxe Percue (Tassa Riscossa) Milano Ferr. Corr.

Spedizione in Abb. Post. Gruppo III/70

ALIMENTATORE PER PROGRAMMATORE DI EPROM

Particolarmente economico, il mini-programmatore di EPROM presentato sul dicembre '89, ha fatto un enorme successo anche perché è stato progettato per trarre il massimo profitto dalle apparecchiature già in possesso dell'utilizzatore: il microcomputer, naturalmente, ma anche gli alimentatori da laboratorio. In questa sede, diamo un utile accessorio: un alimentatore da rete, previsto di funzioni di sicurezza necessarie a proteggere le memorie da programmare. Naturalmente, il costo di questo apparecchio dovrà rimanere contenuto, come quello degli altri elementi della serie.

La programmazione delle memorie EPROM richiede tensioni di alimentazione molto particolari e precise: una tensione di +5 V (generalmente Vcc o Vdd), ma anche una tensione Vpp, con valore diverso da un tipo di memoria all'altro.
Una volta, quasi tutte le EPROM si programmavano in presenza di una Vpp di +25 V. Con il progredire della tecnica, sono comparse versioni che richiedono solo +21 V, oppure addirittura +12,5 V talvolta senza cambio di sigla!
Facciamo notare che la Vpp prescritta deve essere rispettata con una precisione di circa mezzo volt, per non rischiare di perdere la programmazione, di abbreviare la durata utile della EPROM, o addirittura distruggerla. Inoltre, la tensione Vpp non deve essere mai applicata alla memoria in assenza dei +5 V, anche per una sola frazione di secondo: deve essere applicata DOPO ed interrotta PRIMA della tensione di alimentazione principale, per non correre il rischio di distruggere totalmente la EPROM.
Siamo sicuri che i nostri lettori rispetteranno rigorosamente le istruzioni fornite dal software di programmazione, ma rimane sempre possibile una accideentale interruzione della rete: è dunque auspicabile un dispositivo di sicurezza, il quale faccia scendere la Vpp più in fretta della Vcc e che la prima impieghi più tempo a ristabilirsi al ritorno della tensione di rete.
Questo ciclo dovrà pertanto essere rispettato, a prescindere dalle correnti di volta in volta assorbite dalla Vcc e dalla Vpp poiché ci sono differenze sensibili tra le EPROM, a seconda del tipo ed anche della marca.

Uno schema ridotto al minimo

Lo schema elettrico di Figura 1 rispetta i requisiti appena definiti, ricorrendo al minimo indispensabile di componenti. Si potrebbe ovviamente fare meglio ricorrendo, ad esempio, a regolatori integrati asserviti l'uno all'altro, ma sarebbe una soluzione più costosa!
Un rettificatore del tipo "duplicatore di tensione" permette di ricavare da un semplice trasformatore 2 x 9 V (oppure 18 V) la tensione non stabilizzata necessaria, il cui valore è abbastanza elevato (una cinquantina di volt). Questa configurazione permette inoltre di accontentarsi di condensatori di filtro da 2200 μF e 25 V, non molto ingombranti e neppure troppo costosi. Un primo transistor di regolazione in serie BD135, polarizzato mediante uno zener da 5,6 V, fornisce i +5 V necessari con una corrente massima di circa 100 mA. Dovendo realizzare una caduta di tensione di circa 45 V, questo transistor si riscalderà molto: attenzione a non trascurare il problema del

suo raffreddamento, anche a costo di utilizzare un dissipatore termico più grande di quello mostrato sulle foto, che è il minimo indispensabile.
Perché questo transistor deve essere sottoposto ad una tale dissipazione di calore? Semplicemente perché questo tipo di regolatore dispone di una grande riserva di energia, contenuta nei condensatori di filtro: in caso di interruzione della tensione di rete, la tensione di +5 V

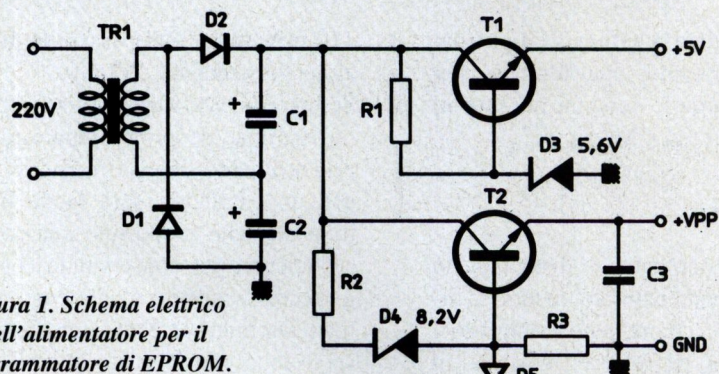



Figura 1. Schema elettrico dell'alimentatore per il programmatore di EPROM.

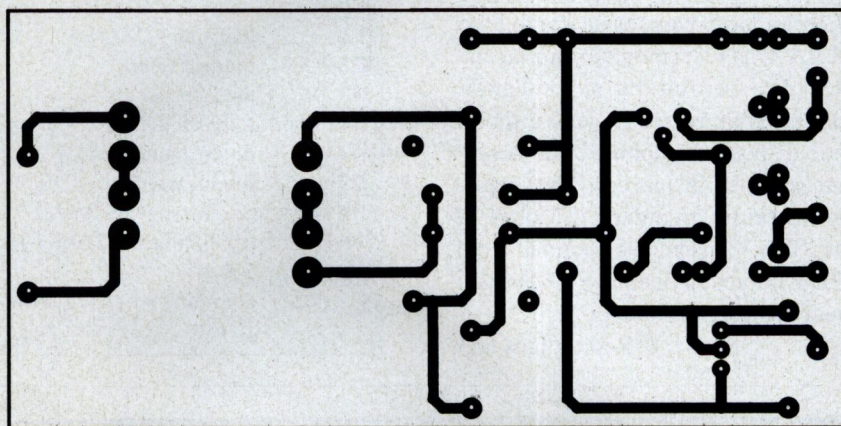



Figura 2. Circuito stampato dell'alimentatore visto dal lato rame in scala unitaria.

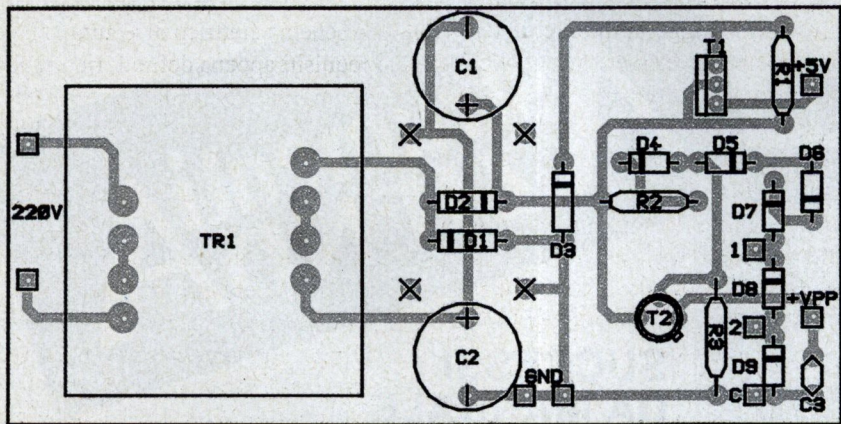



Figura 3. Disposizione dei componenti sulla basetta stampata.

comincerà a scendere soltanto quando la tensione non stabilizzata sarà caduta a circa 7 V.
Durante questo periodo, la Vpp sarà già notevolmente diminuita.
Questa prima linea difensiva è duplicata da una seconda misura di sicurezza: uno zener da 8,2 V, inserito nel ramo superiore del partitore di polarizzazione del transistor regolatore in serie della Vpp (2N1890 o simile, comunque in grado di sopportare almeno 50 V). Grazie ad R3, che collega la base di T2 a massa, quest'ultimo si bloccherà quando la tensione non stabilizzata sarà caduta a 8,2 V e anche un po' prima. Sappiamo però che in questo istante la tensione di +5 V sarà ancora presente!

Inversamente, all'avviamento oppure al ritorno della tensione di rete dopo un'interruzione, la Vpp apparirà soltanto quando i + 5 V si saranno stabilizzati.
La selezione del valore di Vpp fra i tre valori "classici" si effettua cortocircuitando parzialmente un partitore formato da diodi zener e da diodi al silicio, calcolato in modo da avvicinarsi il più possibile alle tensioni raccomandate e dipendere minimamente dalle variazioni della temperatura (ricordiamo infatti che le derive termiche degli zener e dei normali diodi sono opposte).
Naturalmente si potranno scegliere altre combinazioni, a seconda delle esigenze dell'utilizzatore.
Sembra che la semplice manovra di un commutatore sia più comoda e meno pericolosa della regolazione di un potenziometro, effettuata tenendo sotto controllo un voltmetro, ma attenzione a non sbagliare la posizione! Molto spesso le EPROM che richiedono una Vpp diversa da 25 V sono contrassegnate in modo particolare, ma non ci stancheremo mai

di raccomandare la lettura del foglio dati del fabbricante, specialmente per quanto riguarda le versioni più recenti (ad esempio i tipi CMOS).

Costruzione

Il circuito stampato è disegnato in Figura 2 in scala naturale, mentre la disposizione dei componenti, compreso il trasformatore di rete, è mostrata in Figura 3. Le piazzole destinate al trasformatore, possono essere spostate a seconda dell'esatta piedinatura del componente disponibile, a meno che non si preferisca montarlo a parte. Anche T2, come T1, dovrà essere energicamente raffreddato anche se scalda molto di meno. Per la Vpp minima di 12,5 V, si dovrà verificare una caduta di più di 38 V che, con una corrente di 80 mA, corrisponde a circa 3 watt!
Con 25 V si può invece "tirare" fino a 110 mA, senza oltrepassare la dissipazione massima del 2N1890.
Se fossero necessarie correnti superiori (ad esempio, per programmare contemporaneamente diverse memorie collegate in parallelo), sarà facile trovare transistor che sopportino una corrente sufficiente, pur conservando il guadagno (utilizzando eventualmente componenti Darlington). Naturalmente, è indispensabile che il trasformatore sia in grado di erogare la corrente necessaria: nella versione di base, basta un modello da 8 VA. Una volta montato, questo circuito potrà essere inserito nel contenitore del programmatore, dove c'è molto spazio disponibile, oppure in un contenitore separato. In ogni caso, ma soprattutto nel primo, ricordarsi di isolare la parte a 220 V, inserendo eventualmente un interruttore, un fusibile e magari una spia di controllo.

ELENCO DEI COMPONENTI

Tutti i resistori sono da 1 W 5%, salvo diversamente indicato

R1	**resistore da 2,7 kΩ**
R2	**resistore da 1 kΩ**
R3	**resistore da 10 kΩ 0,5 W**
C1-2	**cond. elettr. radiale da 2200 μF 25 Vl**
C3	**cond. in poliestere da 100 nF**
T1	**transistor BD135**
T2	**transistor 2N1890**
D1-2-5-6	**diodi 1N4001**
D3	**diodo zener da 5,6 V**
D4	**diodo zener da 8,2 V**
D7	**diodo zener da 12 V**
D8	**diodo zener da 9,1 V**
D9	**diodo zener da 3,3 V**
TR1	**trasformatore 220/2 x 9 V, 8 VA**
1	**commutatore a tre posizioni**

MOUSE PER C64

Se state ancora usando il vecchio joystick o le paddle, per spostare il cursore lungo lo schermo o per disegnare, è ora di provare il mouse C-1350!
Sul C-128 (o sul C-64 con cartuccia Super Expander), usando il BASIC potrete leggere il mouse mediante la funzione JOY0. Attenzione: soltanto il pulsante di sinistra viene interpretato come pulsante di "sparo". Certamente saprete che questo mouse, analogo come aspetto fisico al mouse Amiga ha due pulsanti: destro e sinistro! Poiché nel manuale non si fa menzione del pulsante destro, abbiamo deciso di "sezionare" il mouse per trovare il motivo.

Piedinatura del mouse

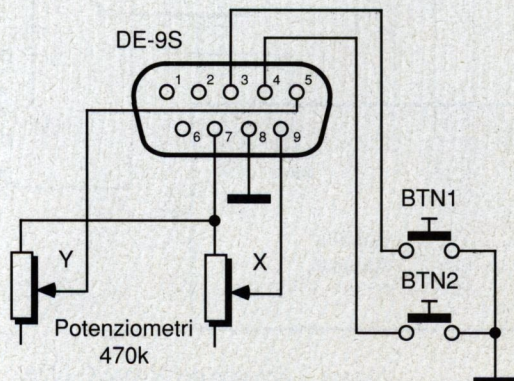
Internamente si tratta di un circuito molto aggiornato: due dischi a codifica ottica, mossi dalla sferetta, generano impulsi in quadratura di fase. Essi vengono decodificati mediante hardware (utilizzando amplificatori operazionali e comparatori) per generare le quattro uscite: UP, DN, LFT e RHT.
Arrivano all'uscita anche i contatti di lavoro dei due pulsanti, il tutto mostrato chiaramente in Figura 1 la quale presenta anche le piedinature di tre tipi di dispositivi d'ingresso per porte di controllo: JOYstick digitale, MOUSE e joySTICK analogico.
Sul mouse C1350, il pulsante di sinistra esce sul piedino 6, come il pulsante di "sparo" JOY, mentre il pulsante destro esce dal piedino 9, come la linea POT X di STICK. POT X e POT Y vengono fatti entrare (READ ONLY) nel convertitore A/D utilizzato per digitalizzare la posizione analogica dei potenziometri. Lo schema del registro $DC00 di CIA 1 mostra la distribuzione dei bit sulle linee logiche digitali di ciascun dispositivo.
Per utilizzare l'uscita del giusto pulsante, cioè quello di sinistra, che effettua una semplice chiusura a massa, è necessario ricorrere ad una combinazione di tecniche analogiche e digitali.
La Figura 2 mostra lo schema di un joystick analogico a due pulsanti Potrebbe anche servire per una pad grafica od altri dispositivi di analoga configurazione fatti in casa. Lo schema mostra graficamente che quanto occorre sulla linea POT X è un potenziometro tra il piedino 9 ed il piedino 7 (+5 V).
La Figura 3 illustra lo schema semplificato del mouse C-1350. Il pulsante sinistro è BTN1 ed il pulsante destro è BTN2. La porta di controllo collega BTN2 al registro $D419. Questo registro permette al microprocessore di leggere la "posizione" ovvero, in questo caso, il "livello logico" della linea POT X, con valori che variano da $00, alla minima resistenza (= livello logico alto), a $FF alla massima resistenza (= livello logico basso). La chiusura dell'interruttore genera un livello logico basso ed allora, per generare un livello logico alto, ci vuole un resistore di "pull-up" tra il piedino 9 (BTN2) ed il piedino 7 (+5 V).

Modifiche al mouse

Effettuando esperimenti con un potenziometro ed un oscilloscopio, abbiamo trovato che il valore di 47 kΩ è quasi giusto per il resistore di pull-up. Non si tratta di un valore critico, ma può variare tra 22 kΩ e 100 kΩ.
Internamente al mouse c'è ancora spazio per un resistore. Ne abbiamo usato uno, molto piccolo, da 1/8 W e 47 kΩ. E' inoltre necessario un cacciavite a croce ed un saldatore a punta fine e basso consumo.
Due viti sul lato inferiore del mouse tengono insieme le due metà dell'involucro, altre due interne fissano il circuito stampato (ed anche quello dei pulsanti) alla metà inferiore del contenitore.
La Figura 4 mostra la serigrafia dei componenti sul circuito stampato. Il cavo del mouse si inserisce in una spina a blocco

Piedino	Joy	Mouse	Stick
1	Su	Su	-
2	Giù	Giù	-
3	Sinistra	Sinistra	Pulsante 1
4	Destra	Destra	Pulsante 2
5	-	-	Potenziometro Y
6	Pulsante 1	Pulsante 1	-
7	+5 V	+5 V	+5 V
8	Massa	Massa	Massa
9	-	Pulsante 2	Potenziometro X

Figura 1. Piedinature delle porte di controllo

Figura 2. Schema di joystick analogico.

	7	6	5	4	3	2	1	0
Joy	Porta 2	Porta 1		BTN1	Destra	Sinistra	Basso	Alto
Stick	Porta 2	Porta 1			BTN2	BTN1		
Mouse	Porta 2	Porta 1		BTN1	Destra	Sinistra	Basso	Alto

CIA 1 - Registro $DC00

montata sulla scheda.
Il codice a colori della spina a blocco è:

Piedino della spina a blocco	Colore	Funzione
1	giallo	RHT
2	arancio	LFT
3	rosso	DN
4	marrone	UP
5	bianco	GND
6	blu	+5 V
7	verde	BTN1
8	nero	BTN2

Saldare i resistori tra i piedini 6 ed 8 di questa spina a blocco, sul lato rame, e rimontare il mouse.

Codice macchina del mouse

Ora che abbiamo un giusto pulsante con due livelli logici distinti, per utilizzarlo è necessario un determinato codice.
Questa è una tipica codifica modularizzata, adattabile alle particolari necessità di ognuno.
Lo scopo è di mantenere le coordinate x ed y e memorizzare lo stato dei pulsanti.

```
;definisci le etichette delle variabili
XPOS.  BYT0          ;salva posizione x
YPOS   .BYT0         ;salva posizione y
BTNS   .BYT0         ;salva stato btn

;subroutine per leggere la Porta di Controllo 2
RDPORT SEI           ;escludi tastiera
       LDA #$C0      ;
       STA $DC02     ;predisponi ddr in lettura
       LDA #$80      ;
       STA $DC00     ;leggi porta di controllo 2
       LDX #$00      ;dai tempo alle
       INX:BNE*-1    ;linee di stabilizzarsi
       LDX $D419     ;leggi POT X
       LDY $D41A     ;leggi POT Y
       LDA #$FF      ;
       STA $DC00     ;resetta porta 2
       RT            ;
```

Potrete ora semplicemente memorizzare la “posizione assoluta”, in base alle coordinate x ed y appena lette, ma le imperfezioni del potenziometro potrebbero causare instabilità dell’immagine sullo schermo. E’ d’obbligo una certa raffinatezza: un algoritmo mobile che stabilizzi i punti “grezzi”.

```
Subroutine per spostare l'algoritmo di media
AVRG    BCS AVRGP    ;se il segno è positivo
AVRGN   EOR #-1      ;se negativo esegui
        ADC #1       ;subt inversione
        LSR          ;permetti il mezzo-peso
        EOR #-1      ;inverti il byte
        CLC          ;per conservare il segno
        ADC #1       ;
        CLC:RTS      ;
AVRGP   LSR          ;permetti il mezzo peso
        CLC:RTS      ;al byte
```

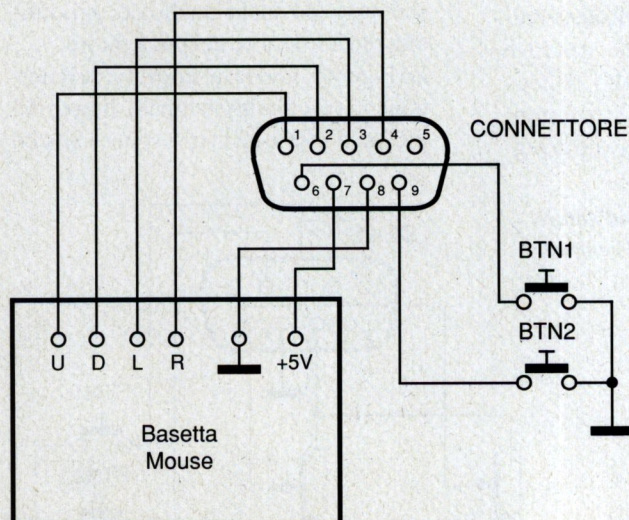



Figura 3. Schema del Mouse C-1350.

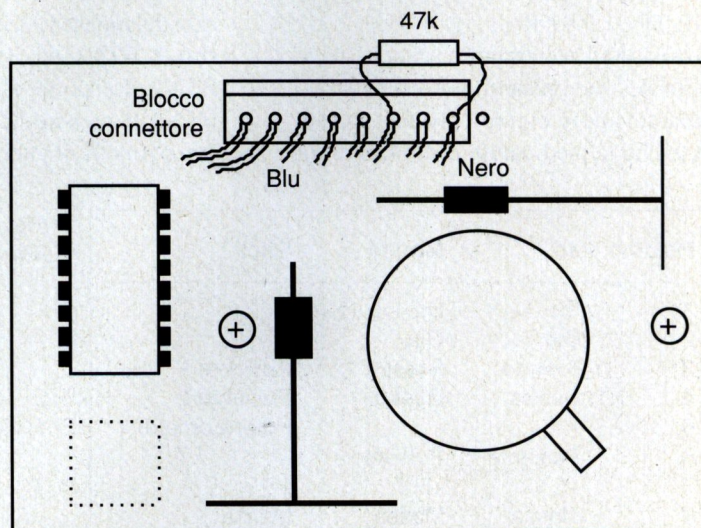



Figura 4. Schema della modifica al mouse.

Raccogliendo il tutto, arriveremo ad una semplice routine che permette di ottenere, senza oscillazioni, la "posizione assoluta" di STICK.

```
;routine principale per leggere STICK
STICK JSR  RDPORT ;
      LDA  $DC00  ;leggi Porta 2
      AND  #50C   ;filtra BTN 1 & 2
      EOR  #$FF   ;inverti la logica
      STA  BTNS   ;salva BTN1 &
                  BTN2
      TXA:SEC     ;inserisci x in
                  XPOS
      SBC  XPOS   ;utilizzando un
                  semplice
      JSR  AVRG   ;algoritmo di
                  media
      ADC  XPOS   ;a spostamen-
                  to
      STA  XPOS   ;aggiorna
                  XPOS
      TYA:SEC     ;inserisci y in
                  YPOS
      SBC  YPOS   ;utilizzando lo
                  stesso
      JSR  AVRG   ;algoritmo
      ADC  YPOS   ;ed aggiorna
                  YPOS
      STA  YPOS   ;

;ora prova i pulsanti ed esci
TEST  LDX  #$FF   ;
      STX  $DC02  ;resetta ddr
      CLI         ;finito con la
                  porta
      LDA  #4     ;prova il bit 2
      BIT  BTNS   ;di BTNS
      BNE  BTN1   ;se premuto
                  BTN1
      ASL         ;prova il bit 3
      BIT  BTNS   ;di BTNS
      BNE  BTN2   ;se premuto
                  BTN2
      RTS         ;esci Z=1 no
                  btns
BTN1  LDA  #-1    ;flag per BTN1
      BYT  $2C    ;salta
BTN2  LDA  #1     ;flag per BTN2
      RTS         ;esci Z=0
```

Il flag Z viene settato se nessun pulsante è stato premuto od altrimenti cancellato. Usare un BEQ per verificare i pulsanti premuti dopo una chiamata JSR STICK. L'accumulatore sa quale pulsante è stato premuto. Usare un BPL od un BMI per controllare quale pulsante è attivo.

Movimenti del mouse

Non è possibile ottenere la "posizione assoluta". Dobbiamo lavorare soltanto con la "posizione relativa", incrementando o decrementando XPOS ed YPOS mentre il mouse si sposta

```
;routine principale per leggere MOUSE
MOUSE JSR RDPORT ;
      LDA $DC00  ;leggi Porta 2
      AND #$10   ;filtra BTN1
      STA BTNS   ;salva il bit 4
      TXA        ;leggi POT X
      BMI LOW    verifica stato
HIGH  LDA #$20   ;setta bit 5
      BYT$2C     ;salta
LOW   LDA #$00   ;cancella bit 5
      ORA BTNS   ;combina i bit
                 4 & 5
      LSR:LSR    ;spostati ai bit
                 2 & 3
      EOR #$FF   ;inverti la logi-
                 ca
      STA BTNS   ;salva BTN1
                 & BTN

;ora abbiamo sia il pulsante sinistro che
quello destro!
      LDA $DC00  ;leggi Porta 2
      AND #$0F   ;direzioni fil-
                 tro
      CMP #$0F   ;qualche mo
                 vimento?
      BEQ EXIT   ;no, finito
      TAX        ;sì, il mouse
                 gira
UP    AND #1     ;verifica salita
      BNE DN     ;
      INC YPOS   ;
DN    TXA        ;
      AND #2     ;verifica di-
                 scesa
      BNE LFT    ;
      DEC YPOS   ;
LFT   TXA        ;
      AND #4     ;verifica sini-
                 stra
      BNE RHT    ;
      DEC XPOS   ;
RHT   TXA        ;
      AND #8     ;verifica de-
                 stra
      BNE EXIT   ;
      INC XPOS   ;
EXIT  JMP TEST   ;prova i pul
                 santi ed esci
```

Usare un BEQ per verificare i pulsanti premuti dopo una chiamata JSR MOUSE. Usare un BPL od un BMI per controllare quale pulsante è attivo.

Il "movimento relativo" della routine MOUSE è molto sensibile. La velocità e l'estensione del movimento, cioè la "sensibilità" del mouse, dipendono da quante volte viene chiamata la routine MOUSE. L'utilizzo delle varie possibilità rimane affidato ai programmatori di mouse!

Conosci l'elettronica?

RISPOSTE AI QUIZ

1. B
2. D
3. C
4. C
5. A
6. B
7. C
8. D
9. D
10. A